

2007P08079

23



Europäisches Patentamt
European Patent Office
Office européen des brevets



Publication number: **0 637 160 A2**

EUROPEAN PATENT APPLICATION

Application number: **94301214.6**

Int. Cl.⁶: **H04M 3/50**

Date of filing: **21.02.94**

Priority: **29.07.93 GB 9315695**

Date of publication of application:
01.02.95 Bulletin 95/05

Designated Contracting States:
DE FR GB

Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

Inventor: **Minakami, Michael Ken**
422 Hogarth Terrace
Sunnyvale,
California 94087 (US)
Inventor: **Hulse, Brian**
65 Hunters Crescent

Romsey,
Hampshire SO51 7UJ (GB)
Inventor: **Cook, Jonathan**
55B Laburnum Grove,
North End
Portsmouth,
Hampshire PO2 0HG (GB)
Inventor: **Pickering, John Brian**
8 St Faith's Road,
St Cross
Winchester,
Hampshire SO23 9QB (GB)

Representative: **Davies, Simon Robert**
IBM
UK Intellectual Property Department
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

Distributed system for call processing.

A first computer workstation 10 includes a voice response unit for interfacing to a telephone network. The first computer workstation is attached by a communications link 18 to a second computer workstation 20, which includes a server to perform a particular voice processing function, such as text to

speech, voice recognition, FAX-back, and so on. For inbound applications the first computer workstation forwards the incoming signal over the communications link to the server on the second computer workstation for real-time processing, whilst for out-bound applications, the reverse process occurs.

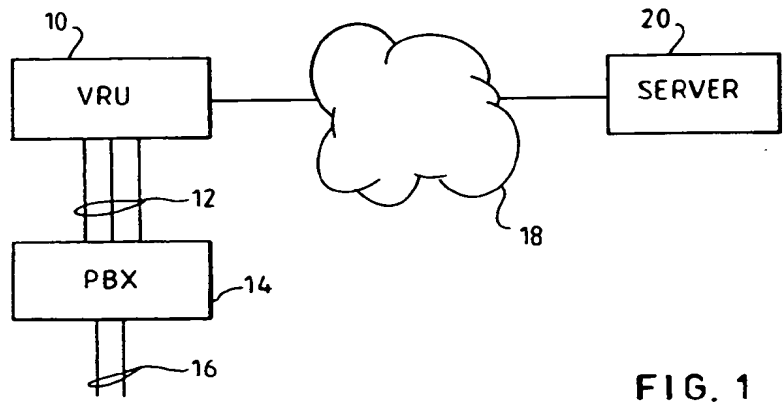


FIG. 1

EP 0 637 160 A2

The present invention relates to a distributed system for interactively processing telephone calls.

Many systems are currently available for automatically providing services or information to callers over the telephone. Often referred to as voice response units (VRUs), such systems have normally relied upon the use of dual tone multiple frequency (DTMF) signals from push-button telephones to obtain input from a caller, and responded largely with pre-recorded voice segments. However, such a system is comparatively limited with respect to the sort of dialogue that can be maintained with the caller, given the restricted range of acceptable inputs, and the need to pre-record any possible responses.

In order to make such communications more natural, and to greatly enhance the flexibility of such systems, it is desirable to equip VRUs with other voice processing technologies, such as voice recognition (to replace the DTMF input), and text to speech (TTS) (to replace pre-recorded voice). There are many varieties of voice recognition that might be considered: for example, voice recognition may operate for discrete words or for continuous speech, and may be speaker dependent, or speaker independent. Recognition vocabularies can range from perhaps 12 words (typically ten digits plus a couple of control words) to many thousands. Likewise, there is a considerable range of TTS technologies available.

Voice recognition and TTS applications tend to be computationally very intensive; for example, full speaker-independent, large vocabulary voice recognition typically requires 100 Mips of digital signal processing power. The requirement becomes even more acute when it is remembered that a VRU may handle perhaps 100 telephone lines simultaneously, lead to a potential maximum processing requirement of 10 Gips. For this reason most commercial systems use specially designed hardware to increase processing speed. These are typically available as Pc adapter cards to be fitted into the VRU.

However, such cards in general must be designed for a particular system, dependent for example on the operating system (DOs or OS/2), computer architecture (ISA or Microchannel), and so on. This greatly restricts the options available to the customer who wishes to incorporate such function into a VRU, since the preferred adapter card may not be compatible with their VRU. Likewise it is difficult to optimise the different components of the system individually. The same problems may also occur even if only software components are involved: for example, a preferred voice mail product may not run under the same operating system as the preferred VRU.

Accordingly, the invention provides a method of interactively processing a telephone call in a distributed system comprising at least two computer workstations connected together by a communications link, in which the first computer is interfaced to the telephone network, said method comprising the steps of:

receiving an incoming telephone signal from the telephone network;

forwarding the incoming telephone signal over the communications link to a server means on the second computer workstation;

processing the incoming telephone signal at the server means and generating a response based upon the telephone signal;

transmitting the response from the server means to the first computer workstation, and processing said call at the first computer workstation in accordance with said response.

The invention also provides a method of interactively processing a telephone call in a distributed system comprising at least two computer workstations connected together by a communications link, in which a first computer workstation is interfaced to the telephone network, said method comprising the steps of:

receiving an incoming telephone signal over said telephone network;

sending a request over the computer network to a server means on the second computer workstation;

generating a telephone signal at the server means based on said request and transmitting the telephone signal to the first computer workstation;

forwarding the generated telephone signal at the first computer workstation out over the telephone network.

The use of a distributed system in which the first computer is primarily responsible for handling the telephone call, and the second computer is primarily responsible for performing the desired technical processing, leads to great flexibility in system design, and so allows optimum technologies to be used in all areas. The server means, which effectively constitutes a remote resource, may be used to provide voice recognition, text-to-speech, voice mail, FAX-back (where a FAX is sent back to the caller), or any other desired facility. These can be divided into inbound and outbound server applications. In the former, such as voice recognition, the server receives an incoming telephone signal (ie the voice of the caller) to process. In the latter, such as text to speech or FAX back, the server produces a telephone signal (synthesised speech or a FAX message) which is transmitted back to the caller. Some applications involve both inbound and outbound processing; for example, a voice mail card will store incoming telephone sig-

nals during the record phase, and but play them out later when the voice mail is examined. The telephone signal may (if desired) be compressed for transmission between the two computer workstations.

It is preferred that the method further comprises the step of a resource controller, located on a computer workstation connected to a said first computer workstation by a computer link, maintaining a list of available server means and current usage thereof, wherein the first computer workstation requests access to the server means from the resource controller, and the resource controller responds to the first computer workstation whether or not the requested server means is available. The resource controller will typically be located on the second computer workstation, but might also be on some third workstation. In some situations it may be convenient to have a single resource control that handles all classes of server (eg voice recognition, text to speech, and so on), but in general it is more convenient to have one resource controller for each class of server.

The preferred initialisation process involves the first computer workstation broadcasting a request to locate the resource controller, and the resource controller responding to the request by sending a message to the first computer workstation specifying the location of the resource controller. An alternative approach would be for the resource controller to send out a broadcast indicating its availability whenever it is brought up.

The invention also provides a distributed system for interactively processing a telephone call, comprising at least two computer workstations connected together by a communications link,

the first computer workstation including interface means for attachment to the telephone network and for receiving an incoming telephone signal over said telephone network; means for forwarding a request over the computer network to a second computer workstation; and means for receiving a telephone signal from the second computer, and forwarding said telephone signal to the interface means for transmission out over the telephone network;

the second computer workstation including server means for generating a telephone signal based on the received request, and means for transmitting the telephone signal to the first computer workstation.

The invention further provides a distributed system for interactively processing a telephone call, comprising at least two computer workstations connected together by a communications link,

the first computer workstation including interface means for attachment to the telephone network and for receiving an incoming telephone sig-

nal over said telephone network; means for forwarding the telephone signal over the communications link to a second computer workstation; and means for receiving a response from the second computer, and for processing the telephone call in accordance with the response;

the second computer workstation including server means for receiving the incoming telephone signal over the communications link and generating a response based upon the incoming signal, and means for transmitting the response to the first computer workstation.

The first and second computer workstations must be linked in a manner providing sufficient bandwidth to support the interactive or real-time processing of the telephone call. A particularly suitable arrangement is for the first and second computer workstations to both be nodes on a local area network (LAN), preferably one having a 16 MBit/s bandwidth. Providing the traffic on the LAN is not unduly heavy, this should allow for example, in a text-to-speech application, synthesised voice to be played out over the LAN without undue delay. Note however, that the situation may become more difficult if the server can support several conversations simultaneously, since in this case the bandwidth requirements are correspondingly increased.

As referred to above, there is preferably a resource controller located on a computer workstation in said local area network, the resource controller maintaining a list of available server means and current usage thereof, and responding to requests from the first computer workstation for access to the server means by notifying the first computer workstation whether or not the requested server means is available. Typically there is one resource controller in the network for each class of available server means.

In one preferred configuration, the LAN includes at least two computer workstations interfaced to the telephone network, and the resource controller (or controllers) manage access from any of the at least two computer workstations to the server means. This has the advantage that multiple computer workstations can share a single server means (eg a single voice recognition unit), which avoids the cost of having to equip each voice response system with such a voice recognition unit.

It should be appreciated that the open architecture of the present invention enables a great variety of possible configurations. For example, within a single LAN a number of voice response units may be supported by a suite of different server devices (text to speech, voice recognition, and so on). The servers may be all be located on one workstation, or may be distributed across two or more machines.

An embodiment of the invention will now be described by way of example with reference to the following drawings:

Figure 1 is a high-level overview of a distributed system according to the invention;

Figure 2 is a detailed diagram of the configuration of the first computer workstation;

Figure 3 is a detailed diagram of the configuration of the second computer workstation;

Figure 4 is a flowchart illustrating the formation of a connection between a voice response unit and a server;

Figure 5 is a flowchart illustrating in more detail the operation of the system with a voice recognition server; and

Figure 6 is a diagram showing a more complicated configuration involving several voice response units.

Figure 1 shows a voice processing system in which a first computer workstation 10, a voice response unit, is connected to a plurality of telephone lines 12 leading to a PBX 14. The telephone lines may be either analogue or digital: in the latter case, there will normally be only a single physical link, with some form of multiplexing. Also shown are trunk lines 16 leading from the exchange into the telephone network. It should be appreciated that in some instances the PBX will not be present, in which case the first computer workstation may be connected to lines leading directly into the telephone network.

The first computer workstation is connected to a second computer workstation 20 by a local area network (LAN) 18. This may for example be a Token Ring network, such as available from IBM Corporation, an Ethernet, or any other form of network that provides sufficient bandwidth and response times to allow interactive real-time processing of a telephone call.

In one particular implementation of the invention, the first computer workstation is a RISC System/6000 running the DirectTalk/6000 software product (both available from IBM Corporation). The second computer workstation is a standard IBM compatible PC with an AT-BUS equipped for example with a VPRO-84 Voice Recognition Card, available from Voice Processing Corporation, Massachusetts, USA. Communications between the first and second computer workstations are carried out using the TCP/IP protocol. This is a conventional protocol based on point to point communications between a port on a first process and a port on a second process (in fact it is possible for the two processes to be on the same machine). Both computer workstations are equipped with suitable adapter cards (as shown in Figures 2 and 3) which allow data to be sent between the two workstations in accordance with this protocol. Such communica-

tions systems are well-known in the art and will not be described in more detail.

Figure 2 illustrates the main components of the first computer workstation running DirectTalk/6000 on an RISC System/6000 under the AIX operating system. This system accepts digital T1 or E1 trunks 12 from the PBX: in the former case, ie in the US, 24 individual channels are multiplexed together into a single trunk, with 8 bits per channel (standard μ -law) and a sampling rate of 8 kHz. The system is attached to the PBX via a digital trunk processor 30 (the 9291 or 9295 cards), plus a digital trunk adapter card 32 located inside the RISC System/6000 itself. These two cards provide the interface to the telephone network, and are responsible eg for demultiplexing incoming calls, and multiplexing outgoing calls. Incoming calls are aggregated into 20 ms blocks of data before being passed on for further processing.

Telephone signals are received from the digital trunk adapter card by a device driver 40, as known in the art, which is responsible for buffering the signals so that they can be collected for processing by other system components. Likewise, the device driver is responsible for receiving outgoing messages from the system and forwarding them to the digital trunk adapter card for transmission onto the telephone network. Data is read from and written to the device driver in accordance with standard programming techniques.

Figure 2 illustrates the main processes running on the RISC System/6000 that are necessary for an understanding of the invention (note the device driver is not an actual process itself, but rather a task running under the operating system kernel). The overall operation of the workstation is supervised by an application program 42, which consists of a set of high level commands. These commands are interpreted by the channel processor 44 which is responsible for allocating resources inside the computer, and establishing connections as required between the various processes. In accordance with the present invention, it is possible for an application to request a resource that is effectively external to the system: in other words the resource is to be supplied by a server on another machine. In this situation, the channel processor requests a custom server process 46 to obtain access to the remote resource. Actual data exchange between the first computer workstation and the server is controlled by a resource processor 48, 50, via a network interface card 60. The number of resource processors, which are started at initialisation by the custom server, can be configured, for example according to the expected system load. During call processing, the resource processor transfers data directly to or from the device driver to allow rapid flow of data between the telephone lines and the

server.

Figure 3 illustrates the processes that are active on the second computer workstation 20 - the structure of this system is in fact very similar to that of the first computer workstation. There is a card 70 which in this particular implementation provides a voice recognition facility, but could be used for example to offer FAX, text to speech, and so on (there is a device driver associated with card 70, but this has not been shown since it is not relevant to an understanding of the present invention). A resource server acts as a front end to the card, allowing on other machines to interact with the card, sending voice data into it and receiving back recognised text. Typically the server and associated card can handle several incoming or outgoing channels simultaneously. The second computer workstation also includes a resource controller 72, which maintains a table of all available servers, together with updated information on their current usage. Finally, the second computer workstation also includes a network interface card 80 to enable communications over the LAN 18.

The sequence of events whereby a remote resource can be utilised is illustrated in Figure 4. The process begins with the CHP sending a request to the custom server to provide access to the resource. In the present implementation, the CHP and the custom server communicate by means of an application programming interface (API) in the CHP. During initialisation, the custom server calls the CHP, effectively informing the CHP of its existence. The custom server then regularly calls the CHP to see if it has any instructions for it (ie whether there is an outstanding request for an external resource).

The custom server now sends a datagram out to the resource controller (a datagram is a special type of message available in AIX: it is used because the resource controller may be supporting more than one voice response unit, and must be open to receiving requests from any machine). The datagram identifies the resource required by the custom server. The resource controller then checks its table of available resources, and their current status, and assuming for the moment that the request can be satisfied, returns a message to the custom server containing information identifying the allocated server (the network address and IP port of the server). The resource controller then updates its table of available resources. The custom server forwards the location information onto the resource processor that will handle the particular server, allowing the resource processor to communicate directly with the server. The custom server and resource controller play no further part in this stage of communications. Finally, when the processing of the server has terminated, the communication be-

tween the server and the resource processor can be concluded, and the custom server and resource controller notified accordingly.

If instead it turns out that the remote resource requested by the custom server is not in fact available, for example the relevant server is already being fully utilised, then clearly the resource controller returns a negative response to the custom server. This response may indicate a suggested time to try again.

Figure 5 shows in more detail the processing associated with a particular in-bound application (voice recognition of discrete digits) once communication between the resource processor and the remote server has been established. A logical connection is made between the resource processor and the device driver (in AIX terminology, a stream is set up between a port on the device driver and a port on the resource processor). This allows the resource processor to read data directly from the device driver (this is much faster than allowing the CHP to do the routing, which is particularly important given the need for real-time processing of the call with minimum delay). Once this connection has been established, the resource processor repeatedly polls the device driver to see if any data has arrived. Whenever it obtains a positive response, it collects the data, forms it into a packet (or packets) together with appropriate control information, and then sends it over the network to the server. The server attempts to identify the digit which has been spoken based on the received signal: if the attempt is unsuccessful, the server must wait for more data. Once a successful recognition has been made, the server can return the spoken digit to the resource processor, which can forward the information to the channel processor (for return to the application), and close down communications between the resource processor and the server.

The initialisation of the distributed system is as follows. When the second computer workstation is initialised, each of the servers or resources notifies the resource controller of its existence, along with the number of ports that can be utilised for call processing. The resource controller can then make the appropriate entries in its resource table for each server. Next, when the first computer workstation is initialised, the custom server broadcasts a message over the LAN in order to locate the resource controller. This produces a response from the resource controller including the address of the machine on which the resource controller resides. Note that if the first computer workstation comes up before the resource controller, so that it does not receive any response to its broadcast message, it simply repeats the message until the resource controller has been started and can reply.

One potential problem with the distributed system so far described is the possibility of packet loss somewhere in the network. To obviate this, whenever a packet containing either data or a command is sent between the resource processor and the server, an acknowledgement is expected. Each packet is stored in a queue after transmission and only deleted after the acknowledgement has been received (the identification of packets is discussed in more detail below). If a specified number of re-transmissions occur, without any acknowledgement being received, it is assumed that the connection between the resource processor and the server has gone down, and appropriate error recovery procedures are initiated.

According to the implemented communication protocol, each packet has a basic 8-byte header, to which further information or data may be appended. The header includes fields identifying the type of packet (discussed in more detail below), various control flags, channel and sequence numbers (again discussed below), information about the length of the packet following the header, and error checking bytes.

The channel id contained in each packet header specifies the channel to which the packet relates. The sequence number (which relates only to packets with that channel id) allows the loss of individual packets to be detected, and a re-transmission request sent if necessary. It also helps ensure that incoming packets are correctly sequenced at the receiving end. The channel id is required bearing in mind that the resource processor and server may be handling several different telephone calls simultaneously. Each call is assigned its own channel, to ensure that traffic on one line does not get confused with traffic on another line (this identification scheme could be extended if the network included several VRUs, as discussed below with reference to Figure 6). A unique channel id of 0 is assigned to communications between the custom server and the resource controller (this channel is not associated with any one particular telephone call).

The different types packets will now be described for the various stages of operation described earlier. The initialisation procedure commences with the custom server sending out an IDENTIFY packet containing its own IP port and address (this is broadcast in datagram mode). The packet specifies a predetermined port number (1500, in the actual implementation), and any resource controller having a port number matching this responds with an AVAIL packet identifying itself and its whereabouts to the custom server. The IDENTIFY packet is resent using a linear or possibly exponential delay if no response is received straight away.

Once the resource controller has been located, the custom server sends a CHANOPEN packet to the IP address and port specified in the AVAIL response to the initial IDENTIFY packet. The purpose of this is simply to confirm the link to the resource controller. The initialisation procedure is completed when the custom server receives an acknowledgement to its CHANOPEN packet from the resource controller.

The CHANOPEN/acknowledgement procedure is also used whenever a new channel is opened between a resource processor and a server to confirm that the channel is operational. Two types of packet, RESEND and RESTART, are provided to handle the situation where packets are lost, as mentioned above. The first of these, RESEND, identifies a particular packet to be resent, for example if examination of the sequence numbers indicates that a packet has not arrived. RESTART by contrast is used where data communications have been more seriously disrupted, and it is decided to recommence the sequence of packet transmission either from the beginning or from some specified packet sequence number.

The CHANREQ packet is sent by the custom server to request a particular service from the resource controller (the packet contains an identification of the desired service). If available, the acknowledgement to this packet will contain the IP port and address of the desired server; if the response is negative, the custom server must decide whether to retry later or abort the attempt. One further aspect is that the acknowledgement may indicate that the server requires application initialisation data to be downloaded from the custom server machine to the server machine: should this be the case, the custom server handles the transmission of the relevant data.

After communications between the server and the resource processor have come to a conclusion, a CHANREL packet is sent by the resource processor. The server should send an acknowledgement back, closes the relevant port, and notifies the resource controller of its updated status. In some applications it may be desirable for the resource processor always to have the same server available: in this case the CHANREL packet is not sent, so that the connection remains open.

Data are transferred using a DATADL packet, which contains free-format data, whether text to be converted into speech, voice signals for recognition, or whatever. In the case of voice recognition, many DATADL packets may be sent (each of which would be acknowledged) until recognition is successfully achieved and a result is available. This result is then returned to the resource processor attached to a RESULT packet. Text-to-speech operates slightly differently, in that both the text sent

to the server, and the synthesised speech returned is transmitted in DATADL packets (again with each packet being acknowledged).

Voice data are transmitted over the LAN using conventional (ie uncompressed) 8-bit μ -law or A-law encoding (depending on the country). Although compression would help increase the rate of communication, it would be necessary for each server and voice response unit in the network to support the same compression algorithm. This would then have the undesirable effect of restricting which VRU could use which server. However, in more sophisticated systems, it may be possible for the resource processor and server to negotiate about compression as part of their initial exchange of messages. If it was determined that both did in fact support the same algorithm this could then be adopted for that communication session. Another possibility is that in future the LAN interface cards will perform compression, although this should be transparent to the sending and receiving applications.

The size of packets used to transmit voice data over the LAN can be adjusted to optimise performance, based on the particular application, network traffic, and so on. For example, text to speech applications may send large packets (4 kbytes) every 0.5 seconds, since this is efficient in terms of LAN utilisation. By contrast, voice recognition may suit a smaller packet size, since not all the data may be required to identify the spoken input (especially if the recognition is limited to distinguishing between just a few possibilities). Note that the current system does not multiplex channels together. Thus if several channels (ie telephone lines) are being handled by the same resource processor and server, each channel will have its own stream and ports at each end. This allows each channel to be closed individually when it is no longer required, and avoids the additional overheads and complexity that would be required to support multiplexing.

Figure 6 shows another distributed system for interactively processing a telephone call. This configuration is more complicated than that of Figure 1, in that the LAN 180 now includes multiple voice response units 130, 140, 150, 160, which are capable of supporting a large number of telephone lines 200. Furthermore, there are also multiple resource controllers, RC1 and RC2 on nodes 110 and 100 respectively, and multiple servers, RS1 on first server machine 110 (offering perhaps TTS), and RS2 and RS3 on a second server machine 120 (offering voice recognition). Such a configuration effectively shares the servers amongst the VRUs, providing a wide range of function that would not be affordable if each VRU needed its own server.

Typically resource controller RC1 would manage allocation of server RS1, while resource con-

troller RC2 would manage allocation of servers RS2 and RS3. In the present implementation, there is a separate custom server in each VRU for each resource controller (although this is entirely dependent on the design of any particular custom server). Thus the VRUs (130-160) would have support two custom servers in order to offer both voice recognition and TTS. The functioning of each of these is analogous to the operation of a single custom server system as previously described.

Claims

1. A method of interactively processing a telephone call in a distributed system comprising at least two computer workstations connected together by a communications link, in which the first computer is interfaced to the telephone network, said method comprising the steps of:
 - receiving an incoming telephone signal from the telephone network;
 - forwarding the incoming telephone signal over the communications link to a server means on the second computer workstation;
 - processing the incoming telephone signal at the server means and generating a response based upon the telephone signal;
 - transmitting the response from the server means to the first computer workstation, and processing said call at the first computer workstation in accordance with said response.
2. The method of claim 1, wherein said server means performs a voice recognition function.
3. A method of interactively processing a telephone call in a distributed system comprising at least two computer workstations connected together by a communications link, in which a first computer workstation is interfaced to the telephone network, said method comprising the steps of:
 - receiving an incoming telephone signal over said telephone network;
 - sending a request over the computer network to a server means on the second computer workstation;
 - generating a telephone signal at the server means based on said request and transmitting the telephone signal to the first computer workstation;
 - forwarding the generated telephone signal at the first computer workstation out over the telephone network.
4. A method as claimed in claim 3, wherein the server means performs a text to speech func-

tion.

5. A method as claimed in any preceding claim, further comprising the step of a resource controller, located on a computer workstation connected to a said first computer workstation by a computer link, maintaining a list of available server means and current usage thereof, wherein the first computer workstation requests access to the server means from the resource controller, and the resource controller responds to the first computer workstation whether or not the requested server means is available. 5
6. A method as claimed in claim 5, wherein the resource controller is located on said second computer workstation. 10
7. A method as claimed in claim 5 or 6, further comprising an initial step of the first computer workstation broadcasting a request to locate the resource controller, and the resource controller responding to the request by sending a message to the first computer workstation specifying the location of the resource controller. 15
8. A distributed system for interactively processing a telephone call, comprising at least two computer workstations connected together by a communications link, 20
 - the first computer workstation including interface means for attachment to the telephone network and for receiving an incoming telephone signal over said telephone network; means for forwarding a request over the computer network to a second computer workstation; and means for receiving a telephone signal from the second computer, and forwarding said telephone signal to the interface means for transmission out over the telephone network; 25
 - the second computer workstation including server means for generating a telephone signal based on the received request, and means for transmitting the telephone signal to the first computer workstation. 30
9. A distributed system for interactively processing a telephone call, comprising at least two computer workstations connected together by a communications link, 35
 - the first computer workstation including interface means for attachment to the telephone network and for receiving an incoming telephone signal over said telephone network; means for forwarding the telephone signal over the communications link to a second computer workstation; and means for receiving a response from the second computer, and for processing the telephone call in accordance with the response; 40

workstation; and means for receiving a response from the second computer, and for processing the telephone call in accordance with the response;

the second computer workstation including server means for receiving the incoming telephone signal over the communications link and generating a response based upon the incoming signal, and means for transmitting the response to the first computer workstation. 45

10. The distributed system of claim 8 or 9, wherein the first and second computer workstations are both nodes on a local area network (LAN), and said communications link is provided by the LAN. 50
11. The distributed system of claim 10, further including a resource controller located on a computer workstation in said local area network, the resource controller maintaining a list of available server means and current usage thereof, and responding to requests from the first computer workstation for access to the server means by notifying the first computer workstation whether or not the requested server means is available. 55
12. The distributed system of claim 11, wherein the LAN includes at least two computer workstations interfaced to the telephone network, and the resource controllers controls access from any of the at least two computer workstations to the server means. 60

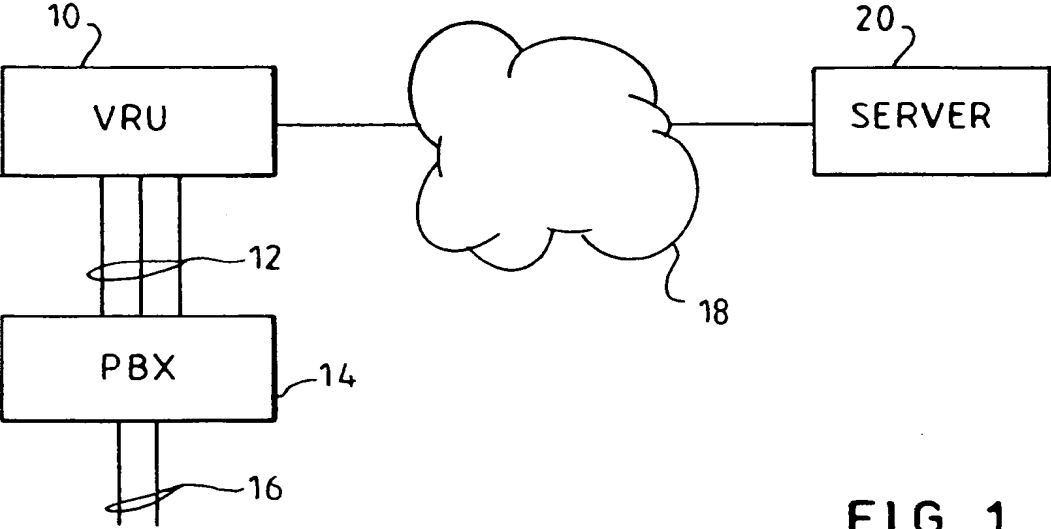


FIG. 1

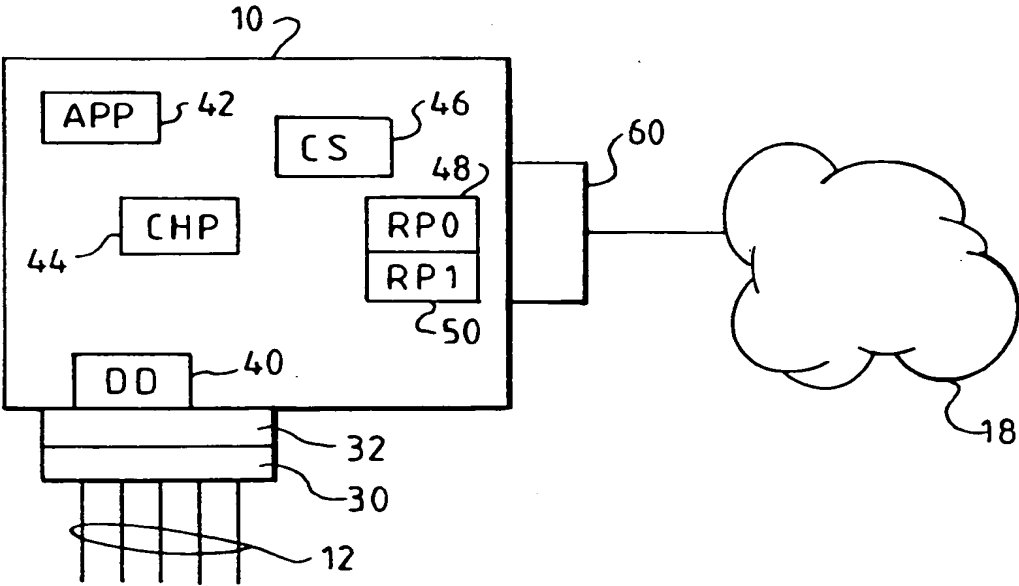


FIG. 2

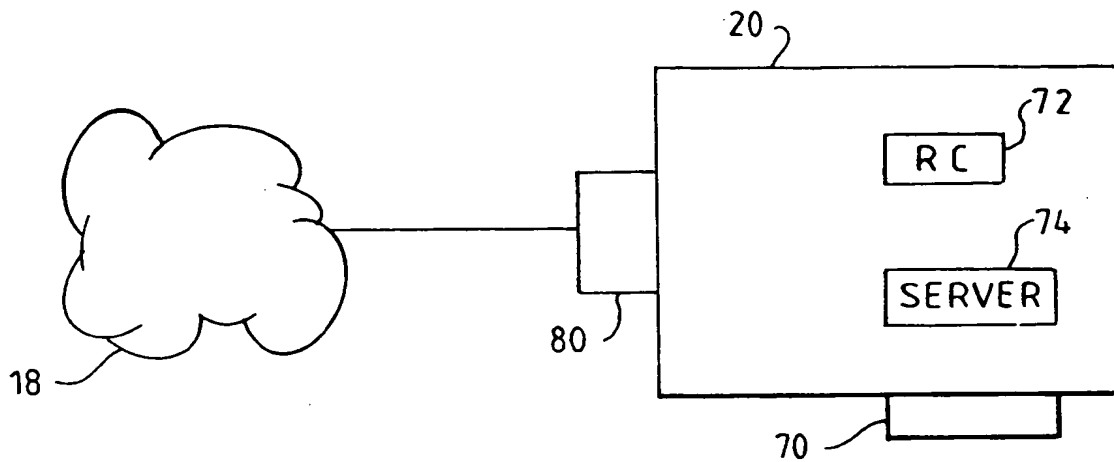


FIG. 3

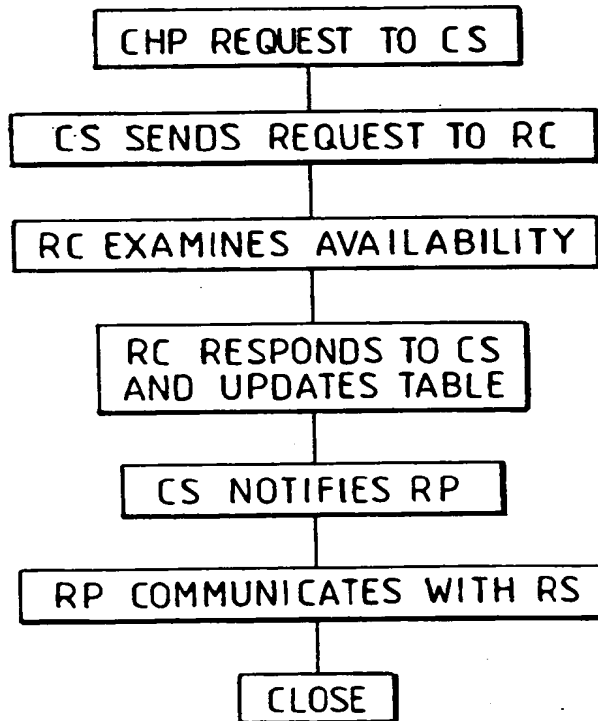


FIG. 4

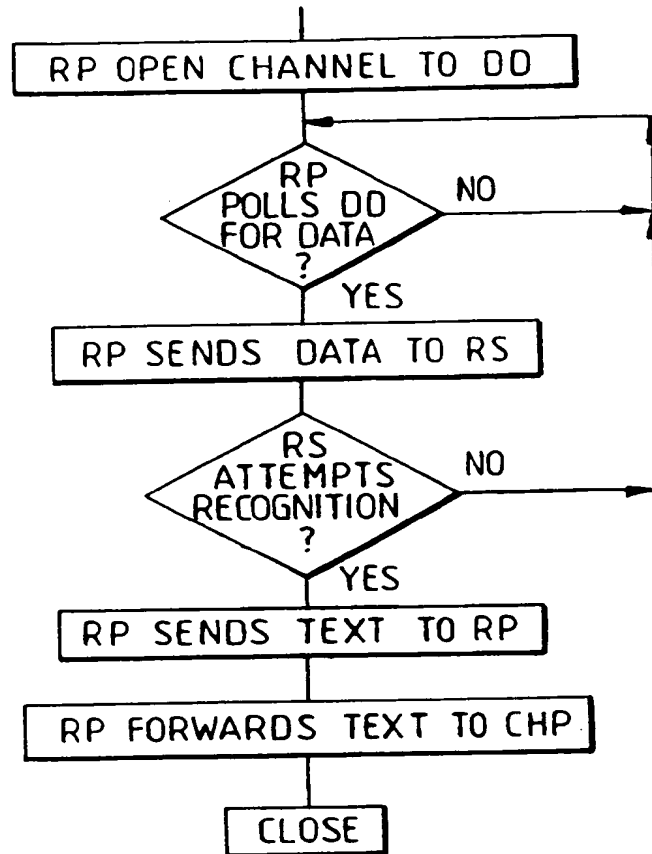


FIG. 5

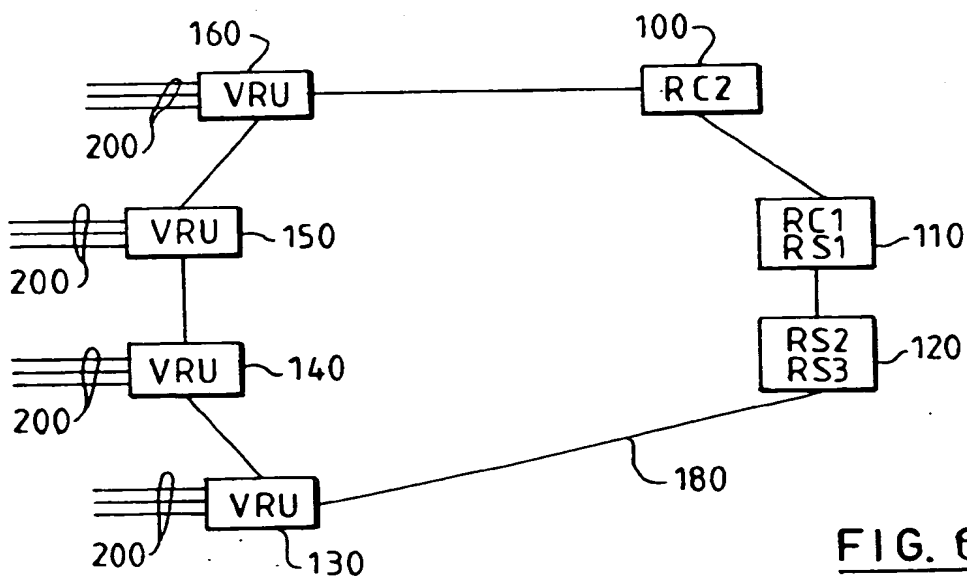
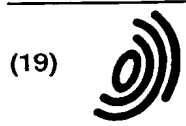


FIG. 6



(19)

Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 637 160 A3

(12)

EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
08.09.1999 Bulletin 1999/36

(51) Int. Cl.⁶: H04M 3/50, G06F 3/16

(43) Date of publication A2:
01.02.1995 Bulletin 1995/05

(21) Application number: 94301214.6

(22) Date of filing: 21.02.1994

(84) Designated Contracting States:
DE FR GB

(30) Priority: 29.07.1993 GB 9315695

(71) Applicant:
International Business Machines
Corporation
Armonk, N.Y. 10504 (US)

(72) Inventors:
• Minakami, Michael Ken
Sunnyvale, California 94087 (US)

- Hulse, Brian
Romsey, Hampshire SO51 7UJ (GB)
- Cook, Jonathan
Portsmouth, Hampshire PO2 0HG (GB)
- Pickering, John Brian
Winchester, Hampshire SO23 9QB (GB)

(74) Representative:
Davies, Simon Robert
I B M
UK Intellectual Property Department
Hursley Park
Winchester, Hampshire SO21 2JN (GB)

(54) Distributed system for call processing

(57) A first computer workstation 10 includes a voice response unit for interfacing to a telephone network. The first computer workstation is attached by a communications link 18 to a second computer workstation 20, which includes a server to perform a particular voice processing function, such as text to speech, voice recognition, FAX-back, and so on. For inbound applica-

tions the first computer workstation forwards the incoming signal over the communications link to the server on the second computer workstation for real-time processing, whilst for outbound applications, the reverse process occurs.

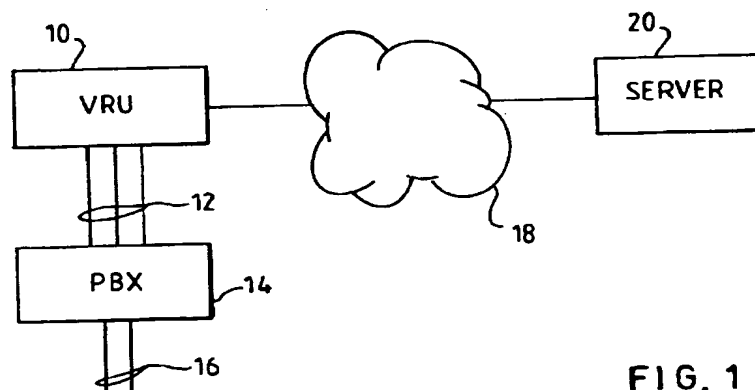


FIG. 1

EP 0 637 160 A3



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 30 1214

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	EP 0 529 915 A (IBM) 3 March 1993 (1993-03-03) * abstract; figure 1 * * column 3, line 41 - column 5, line 25 * ---	1,5,7, 9-12	H04M3/50 G06F3/16
X	US 5 003 574 A (DENQ DONG L ET AL) 26 March 1991 (1991-03-26) * abstract; figure 1 * * column 3, line 37 - line 59 * * column 4, line 26 - line 27 * * column 5, line 56 - column 7, line 26 * ---	1-6,9-12	
X	WO 92 06548 A (US ADVANCED NETWORK INC) 16 April 1992 (1992-04-16) * abstract; figures 1-4 * * page 4, line 30 - page 5, line 30 * * page 10, line 8 - line 18 * * figures 5-9 * ---	1,9,10 11,12	
Y	US 4 959 854 A (CAVE ELLIS K ET AL) 25 September 1990 (1990-09-25) * abstract; figures 3,4 * ---	11,12	TECHNICAL FIELDS SEARCHED (Int.Cl.6)
X	EP 0 173 272 A (COGNITRONICS CORP) 5 March 1986 (1986-03-05) * abstract; figure 1 * * page 11, line 11 - line 20 * ---	1,9,10	H04M G06F
X	GB 2 032 153 A (TOKYO SHIBAURA ELECTRIC CO; KEIHIN ELECTRIC EXPRESS) 30 April 1980 (1980-04-30) * abstract; figure 1 * * page 3, line 4 - line 40 * ---	9,10	
P,X	US 5 283 731 A (LALONDE JAMES E ET AL) 1 February 1994 (1994-02-01) * abstract * * column 3, line 40 - column 6, line 41 * -----	1,9,10	
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 14 July 1999	Examiner Willems, B
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO.

EP 94 30 1214

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

14-07-1999

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0529915	A	03-03-1993	US 5283888 A	01-02-1994
			DE 69225851 D	16-07-1998
			DE 69225851 T	18-02-1999
			JP 2004097 C	20-12-1995
			JP 5257849 A	08-10-1993
			JP 7036174 B	19-04-1995

US 5003574	A	26-03-1991	CA 2010481 A,C	30-09-1990

WO 9206548	A	16-04-1992	US 5113430 A	12-05-1992
			AT 166511 T	15-06-1998
			AU 8950091 A	28-04-1992
			CA 2069009 A	02-04-1992
			DE 69129449 D	25-06-1998
			DE 69129449 T	18-02-1999
			EP 0503059 A	16-09-1992
			US 5487103 A	23-01-1996
			US 5581607 A	03-12-1996
			US 5619556 A	08-04-1997
			US 5692034 A	25-11-1997
			US 5652785 A	29-07-1997
			US 5317627 A	31-05-1994
			US 5335266 A	02-08-1994
			US 5355403 A	11-10-1994
			US 5835572 A	10-11-1998

US 4959854	A	25-09-1990	NONE	

EP 0173272	A	05-03-1986	CA 1239717 A	26-07-1988
			US 4782509 A	01-11-1988
			US 5052038 A	24-09-1991

GB 2032153	A	30-04-1980	JP 55053767 A	19-04-1980
			US 4255619 A	10-03-1981

US 5283731	A	01-02-1994	AU 4405393 A	19-07-1994
			AU 6015594 A	19-07-1994
			WO 9415428 A	07-07-1994
			WO 9415294 A	07-07-1994

ORM P0439

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

THIS PAGE BLANK (USPTO)